THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

DATS 6312 : NATURAL LANGUAGE PROCESSING

FINAL PROJECT REPORT

ON

CUSTOMER COMPLAINT ANAYSIS AND PREDICTION SYSTEM

Leveraging NLP To Enhance Customer Experience

By:

BHUMIKA MALLIKARJUN HORAPET - G29228963

SWATHI MURALI SRINIVASAN – G29020307

MANOJ PADALA – G35417898

SWATHI KR

Group - 7

1. INTRODUCTION

In today's fast-paced digital world, businesses increasingly rely on consumer feedback to assess customer satisfaction, identify product or service issues, and improve overall operational efficiency. The consumer dataset used in this project provides a rich and diverse collection of consumer-related data, capturing interactions between consumers and companies across various industries. This dataset encompasses information such as product details, complaint narratives, company responses, and even consumer consent, offering a holistic view of customer experiences.

The project focuses on leveraging advanced Natural Language Processing (NLP) and Machine Learning techniques to process and derive meaningful insights from this dataset. The four primary tasks that were undertaken are:

- **Complaint Classification**: Classifying consumer complaints into relevant categories based on the type of issue reported, enabling companies to effectively address different customer concerns.
- **Summarization**: Automating the process of summarizing lengthy consumer narratives into concise, informative summaries. This allows businesses to quickly grasp the key points of customer feedback without sifting through extensive text.
- **Sentiment Analysis**: Evaluating the sentiment expressed in consumer narratives to understand the emotional tone behind customer complaints or feedback. This helps in determining customer satisfaction levels and identifying areas for improvement.
- **Company Response Classification**: Assessing the quality of company responses to consumer complaints, focusing on aspects like timeliness and relevance. This task is crucial for evaluating the effectiveness of customer service practices and improving response strategies.

Through this project, we aim to build a robust framework that automates the analysis of consumer feedback, providing businesses with actionable insights to enhance customer satisfaction and improve service quality. By processing large-scale consumer data, companies can gain a deeper understanding of their customer base, quickly identify recurring issues, and optimize their responses to customer complaints.

By leveraging cutting-edge machine learning models and NLP algorithms, this project aims to streamline consumer feedback analysis, making it more efficient, scalable, and insightful for decision-makers.

Consumer Financial Protection Bureau

2. DESCRIPTION OF THE DATA SET

The dataset used for this project is a comprehensive consumer feedback dataset, which includes a variety of features that describe the complaint process and its resolution. The dataset consists of consumer complaints, categorized into different product types and issues, and includes additional attributes about the company responses. The following columns are included in the dataset:

- Date received: The date when the complaint was received.
- Product: The product type being complained about.
- Sub-product: A more specific classification of the product.
- Issue: The main issue reported by the consumer.
- Sub-issue: More specific details about the issue.
- Consumer complaint narrative: The detailed complaint text provided by the consumer.
- Company public response: The company's public response to the complaint.
- Company: The company handling the complaint.
- State: The state of the consumer filing the complaint.
- ZIP code: The ZIP code of the consumer.
- Consumer consent provided?: Whether the consumer consented to having their complaint data processed.
- Submitted via: The method through which the complaint was submitted (e.g., online, phone).
- Date sent to company: The date when the complaint was sent to the company.
- Company response to consumer: The company's response to the consumer.
- Timely response: Whether the company responded in a timely manner.
- Complaint ID: A unique identifier for each complaint.
- Complaint Length: Length of the complaint text.
- Processed Narrative: A pre-processed version of the complaint text(Generated by us).
- Word Count: The total number of words in the complaint.
- Word Count Range: Range for the length of the complaint.
- Summary: A brief summary of the complaint.(Generated by us)

The data consists of multiple complaint categories, and the primary challenge lies in classifying complaints accurately while handling imbalanced classes and diverse complaint structures. This dataset serves as the foundation for our complaint classification, summarization, sentiment analysis, and company response classification tasks.

Rows: 2199541

Size of Data: 4.05GB

Dates: 01/01/2018 to 11/11/2024

3.1 Complaint Classification

We implemented and evaluated several transformer-based models for the classification of consumer complaints. The models we used include XLM-RoBERTa, DistilBERT, RoBERTa, and ALBERT. These models were pre-trained on large text corpora and fine-tuned for the complaint classification task. We used the Hugging Face Transformers library for all models, which provides a straightforward interface for working with state-of-the-art NLP models.

Model Selection and Initial Training

After preprocessing, we worked on implementing and training multiple transformer-based models to compare their performance on the complaint classification task. The models I used were:

- **DistilBERT** (base version)
- **RoBERTa**
- **XLM-RoBERTa**
- **ALBERT**

For each of these models, the training involved the following steps:

Loading Pretrained Models: We utilized the Hugging Face Transformers library to load the pre-trained models. These models had already been trained on vast text corpora and were fine-tuned for the classification task on the consumer complaint dataset.

Loss Function and Optimizer: Used cross-entropy loss for the classification task and AdamW optimizer for training the models. The AdamW optimizer is well-suited for training transformer models, and it adjusts the learning rate during training.

Class Weights: Given the class imbalance in the dataset, and calculated class weights and incorporated them into the loss function to give more importance to underrepresented classes during training.

Training Process: The model was trained using early stopping, which ensures that training halts once the validation loss no longer improves, preventing overfitting. The training loop included these steps:

Differences Between Initial DistilBERT and Final Updated DistilBERT

In the first iteration, the DistilBERT model showed reasonable performance, but there were noticeable opportunities for improvement. The final updated DistilBERT model incorporated several key changes:

Improved Early Stopping Criteria: In the initial model, early stopping was based solely on the validation loss. In the updated model, We have added a more robust early stopping criterion, which saved the model whenever there was a significant improvement in validation performance.

Fine-Tuning the Model: The most significant update was the fine-tuning of the DistilBERT model. After evaluating its initial performance, Reduced the learning rate, retrained the model for more epochs, and achieved a substantial improvement in performance.

Model Performance Comparison

First, let’s compare the performance of all four models on the test dataset. The models were evaluated based on accuracy, macro F1-score, weighted average F1-score, top-3 accuracy, and validation loss.

Model	Accuracy	Macro Avg F1-Score	Weighted Avg F1-Score	Top-3 Accuracy	Validation Loss
DistilBERT (Initial)	58.25%	0.16	0.34	0.58	2.7234
RoBERTa	56.19%	0.17	0.32	0.56	2.8800
ALBERT	37.91%	0.06	0.16	0.37	3.5996
XLM-RoBERTa	30.00%	0.15	0.30	0.52	3.0853
DistilBERT (Fine-Tuned)	67.71%	0.19	0.38	0.67	2.8119

Training Loss and Fine-Tuning Results

The initial DistilBERT model showed a slower convergence in training, with a higher validation loss indicating that it was not performing as well on unseen data.
After fine-tuning, the DistilBERT (Fine-Tuned) model showed a marked improvement in both training loss and validation loss, confirming that the adjustments made during fine-tuning led to better generalization and reduced overfitting.

Model Evaluation Metrics Comparison

Metric	DistilBERT (Initial)	DistilBERT (Fine-Tuned)
Accuracy	58.25%	67.71%
Macro Avg F1-Score	0.16	0.19
Weighted Avg F1-Score	0.34	0.38
Top 3 Accuracy	0.58	0.67
Validation Loss	2.7234	2.8119

Key Learnings

DistilBERT was the most efficient and effective model for this particular task. Despite being a smaller version of BERT, it provided competitive results in terms of both speed and accuracy, making it well-suited for production environments where computational efficiency is crucial.

Fine-tuning played a critical role in improving DistilBERT's performance. The increase in accuracy and top-3 accuracy after fine-tuning demonstrated the importance of model refinement to achieve better generalization and minimize overfitting.

Class Imbalance was a challenge throughout the project, and the use of class weights helped mitigate this issue. The class weights allowed the model to focus more on the minority classes, preventing the model from being biased toward the majority class.

Model Selection: The comparison of multiple transformer-based models revealed that not all models were equally effective for this task. For example, XLM-RoBERTa and ALBERT showed poor results, indicating that choosing the right model for the task is essential. DistilBERT's balance of performance and efficiency made it the optimal choice.

Conclusion

In conclusion, the project demonstrated that DistilBERT is an effective and efficient model for consumer complaint classification. Through a series of experiments, we showed that fine-tuning the pre-trained DistilBERT model led to significant improvements in performance, achieving an accuracy of 67.71% and top-3 accuracy of 0.67.

The lessons learned from this project highlight the importance of careful model selection, hyperparameter tuning, and handling class imbalance in text classification tasks. The improvements made to DistilBERT throughout the project show the potential for fine-tuning pre-trained models to achieve better generalization and performance on real-world datasets.

Future improvements can be made by exploring different models, advanced techniques for handling class imbalance, and fine-tuning the model with additional data augmentation strategies. Despite the challenges faced, the results of this project serve as a strong foundation for future work in complaint classification and other NLP applications.

3.2. Summarization

Preprocessing: I preprocessed the data by removing white spaces, newlines, converting the text to lowercase, eliminating punctuation, and removing legal references and certain XXXX patterns.

Since the dataset was large and summarizing all the data would take a considerable amount of time, I filtered it down to 67,000 rows. The maximum length for summaries was set to 120 words, and the minimum length was set to 20 words.

Summarization Models

T5-Small:

The T5-Small model leverages the "text-to-text" framework, where all NLP tasks are converted into a text generation problem. For summarization, the input is a long complaint narrative, and the model generates a concise summary.

In this project, T5-Small was initially used to summarize 67,000 complaint narratives. While it achieved basic summarization, it struggled with longer and more complex inputs, resulting in a ROUGE score of 0.21. This limitation led to the exploration of a more advanced model.

Time taken to summarise: 9 hours

Evaluation Metrics:

ROUGE Score: 0.21

Since the score was very bad, We switched to the facebook/bart-large-cnn model, which took approximately 13 hours to summarize the data.

BART-Large-CNN:

BART is a denoising autoencoder that excels at text generation and summarization. For summarization tasks, BART fine-tunes on large-scale datasets like CNN/Daily Mail, enabling it to produce coherent and concise outputs.

By using BART-Large-CNN, the summarization improved significantly, achieving an average ROUGE score of 0.71. This model effectively handled lengthy and intricate complaint narratives, making it ideal for this use case.

Evaluation Metrics for BART model:

Average ROUGE Score: 0.71

ROUGE-1: 0.7361

ROUGE-2: 0.7020

ROUGE-L: 0.6813

Time taken: 13hours

The model was saved to the local and used for finetuning the model for classification of Company Response to Consumer.

3.3. Sentiment Analysis

Initial Model Comparison:

The first step was to conduct a comprehensive model comparison for sentiment analysis using customer complaints data. The primary goal was to assess the performance of various pre-trained transformer models in classifying sentiments into three categories: positive (2), neutral (1), and negative (0). The "Processed Narrative" column which was cleaned in the previous steps was used, and each narrative was labelled with its respective sentiment using VADER based on sentiment polarity.

Each model's training and validation were conducted using a split of the dataset (60% training, 20% validation, 20% test).

Models Compared

The following transformer-based architectures were selected for evaluation:

1. **BERT (base-uncased)**: A general-purpose transformer model designed for a variety of NLP tasks.
2. **RoBERTa (base)**: An optimized version of BERT that focuses on robust language understanding.
3. **DistilBERT (base-uncased)**: A smaller, faster variant of BERT designed for efficiency without sacrificing significant accuracy.
4. **T5 (base)**: A sequence-to-sequence transformer model adapted for classification tasks

Evaluation Metrics

The models were evaluated on:

- **Accuracy**: The proportion of correct predictions over the total number of predictions.
- **Precision**: The ratio of true positive predictions to all positive predictions made.
- **Recall**: The ability of the model to identify all relevant instances in the data.
- **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure.

Results:

Model	Accuracy	Precision	Recall	F1-Score
BERT (base-uncased)	0.9028	0.84	0.82	0.83
RoBERTa (base)	0.8920	0.88	0.80	0.83
DistilBERT (base-uncased)	0.8942	0.84	0.81	0.82
T5 (base)	0.8418	0.81	0.68	0.72

Analysis

- BERT (base-uncased) emerged as the best-performing model in terms of accuracy and F1-score. Its balanced precision and recall values suggest it handled all sentiment categories effectively.
- RoBERTa (base) demonstrated high precision, making it adept at reducing false positives, but its recall was slightly lower, indicating some missed predictions.
- DistilBERT (base-uncased) showed competitive performance, making it a viable alternative when computational efficiency is a priority.
- T5 (base) lagged behind the other models, particularly in recall, suggesting it struggled to generalize effectively for this task.

Based on this evaluation, BERT (base-uncased) was selected as the best model for further optimization and application in this project. Its superior accuracy and F1-score make it well-suited for the demands of sentiment analysis on customer complaints.

Model Implementation Details:

The sentiment analysis task was implemented using a fine-tuned BERT (bert-base-uncased) model for classifying customer complaints into three sentiment categories: Negative (0), Neutral (1), and Positive (2). The model architecture leverages BERT's pre-trained knowledge while adapting it specifically for sentiment classification.

Data Preparation

- The dataset consists of consumer complaints with processed narrative text
- A 5-fold stratified cross-validation approach was implemented to ensure robust evaluation
- Class imbalance was addressed using WeightedRandomSampler to ensure balanced training

Model Configuration

- Base Model: BERT (bert-base-uncased)
- Maximum Sequence Length: 128 tokens
- Batch Size: 32

- Learning Rate: 2e-5
- Optimizer: AdamW with linear learning rate scheduler
- Training Epochs: 5

Training Strategy

The training process incorporated several optimization techniques:

- Linear learning rate scheduling with warmup
- Gradient clipping to prevent exploding gradients
- Weighted sampling to handle class imbalance
- Stratified 5-fold cross-validation for robust evaluation

This comprehensive approach resulted in a highly effective sentiment analysis model suitable for processing customer complaints with high accuracy and reliability.

Performance Analysis

The model achieved exceptional performance across all sentiment classes:

Class-wise Metrics

	Precision	**Recall**	**F1-Score**	**Support**
Negative	**0.** 97	**0.**97	**0.** 97	31,258
Neutral	**0.94**	**0.92**	**0.93**	3,342
Positive	**0.97**	**0.97**	**0.97**	33,048

Negative Sentiment: Precision: 0.97, Recall: 0.97, F1-Score: 0.97 (Support: 31,258)

Neutral Sentiment: Precision: 0.94, Recall: 0.92, F1-Score: 0.93 (Support: 3,342)

Positive Sentiment: Precision: 0.97, Recall: 0.97, F1-Score: 0.97 (Support: 33,048)

Overall Performance

- Accuracy: 0.97
- Macro Average: 0.96
- Weighted Average: 0.97

Key Observations:

- The model demonstrates robust performance with several notable characteristics:
- Consistent performance across negative and positive classes
- Slightly lower but still strong performance on neutral class
- High support numbers indicate a substantial dataset
- Balanced precision and recall scores suggest stable classification

3.4. Company Response Classification with Bert-base-uncased

Models Overview

Two models were developed using the bert-base-uncased architecture, known for its effectiveness in handling language-based data through techniques like Masked Language Modeling (MLM) and Next Sentence Prediction (NSP).

Model1: Finetuned Bert Base Uncased Model with Summary

Since we limit the input to 512 tokens for the model, important tokens that appear at the end of a narrative may be missed, potentially leading to incorrect predictions. To address this issue, we summarized the narratives and trained the model on the summarized data.

For predicting the "Company Response to Consumer," we have used the columns "Company," "Issue," "Product," and "Summary" to fine-tune our model1.

Model2: Fine-tuned BERT Base Uncased Model with Processed Narrative and Large Data features used for finetuning are "Company," "Issue," "Product," and "Complaint Narrative"

Data Description

The dataset consists of 67,000 records for Model 1 and 222,437 records for Model 2, drawn from customer complaints detailing issues with various products and companies.

Data Distribution

Model 1 Dataset:

- Closed with explanation: 54,065
- Closed with non-monetary relief: 9,700
- Closed with monetary relief: 3,023
- Closed: 689
- Untimely response: 47

Model 2 Dataset:

- Closed with explanation: 100,000
- Closed with non-monetary relief: 100,000
- Closed with monetary relief: 12,703
- Closed: 1,378
- Untimely response: 2,111

Hyperparameter Tuning for Model Fine-Tuning

In the process of fine-tuning our models, we utilized a set of specific hyperparameters that were carefully chosen based on preliminary testing to optimize performance. Below, we outline the configuration used during training and validation phases:

Data Loaders

- **Batch Size:** A batch size of 16 was used for both training and validation data loaders to balance the trade-off between training speed and memory usage. The choice of batch size affects how quickly the model learns and can have a significant impact on the final model accuracy.

- **Training DataLoader:** Utilized a RandomSampler to ensure that the training data is shuffled for each epoch, helping to reduce model overfitting and improving generalization.
- **Validation DataLoader:** Employed a SequentialSampler which does not shuffle the data, appropriate for validation purposes where we need consistent data ordering to accurately evaluate model performance.

Optimizer and Scheduler

- **Optimizer:** The AdamW optimizer was chosen with a learning rate of 2e-5 and an epsilon value of 1e-8. AdamW is an adaptive learning rate method that has been shown to be effective in deep learning scenarios, particularly in handling sparse gradients on noisy problems.
- **Scheduler:** A linear scheduler with no warmup steps was used. The total number of training steps was set to the product of the number of epochs and the length of the training data loader. This scheduler adjusts the learning rate linearly towards zero, following the initial constant learning rate phase.

Training Process

- **Epochs:** The model was trained over 3 epochs. This number of epochs was chosen to prevent overfitting while ensuring sufficient exposure to the training data.
- **Functionality:** The train_and_evaluate function encapsulates the training loop and evaluation on the validation set. This function integrates the model training with periodic evaluation, which helps in monitoring model performance and making adjustments if needed.

This configuration of hyperparameters and training strategy was pivotal in enhancing the model's ability to generalize to new data while avoiding overfitting.

Evaluation metrics of Model1:

Imbalance Handling: Addressed using class weights.

Validation F1-Score: 0.82

Validation Accuracy: 0.84

Validation Loss: 0.454

Evaluation metrics of Model2:

Validation F1-Score: 0.81

Validation Accuracy: 0.77

Validation Loss: 0.67

3.5 Company Response Classification Using Roberta

As part of the project, I personally contributed by designing and implementing the workflow for fine-tuning the **RoBERTa transformer model** to classify customer complaints based on their resolution status. This work involved the end-to-end development of a robust pipeline, ensuring accurate and scalable results. Below is a detailed account of my contributions:

- **Data Preparation**
    - Extracted and preprocessed complaint features, including **Summary**, **Product**, **Issue**, and **Company**, to create a unified textual representation.
    - Encoded the target labels (**Company response to consumer**) using **LabelEncoder**, facilitating multi-class classification.
- **Tokenization**
    - Utilized the **RobertaTokenizer** from Hugging Face to tokenize and encode the textual data. Ensured optimal sequence formatting with padding, truncation, and special tokens.
    - Maintained a sequence length of 256 tokens, balancing computational efficiency and information retention.
- **Model Fine-Tuning**
    - Adapted the pre-trained **RoBERTa-base model** for classification by adding a classification head and fine-tuning it on the dataset.
    - Implemented the **AdamW optimizer** with learning rate scheduling for effective model optimization.
    - Developed data loaders to facilitate efficient training with batched input.
- **Training and Evaluation**
    - Built a training loop using **CrossEntropyLoss** to fine-tune the model over multiple epochs.
    - Monitored performance through weighted **F1-scores** and validation loss, ensuring the model effectively handled imbalanced classes.
- **Model Deployment**
  Saved the trained model and tokenizer for future use, enabling seamless inference and scalability.
    - Generated predictions on unseen data to evaluate real-world performance and applicability.

Evaluation Metrics:

Achieved high weighted F1-scores and low validation loss, demonstrating the model's robustness in categorizing consumer complaints.

This contribution provided a critical component to the overall project by enabling automated classification of customer complaints, which forms the foundation for analyzing company responses and consumer satisfaction trends.

For company response classification, the goal was to assess whether a company's response to a complaint was timely and appropriate. RoBERTa was used for this task.

Rows used: 67k

Columns used: Company, Issue, Category and Summary,

Test train split: 80:20

Epochs: 4

Model	Accuracy	Macro F1-Score	Weighted F1-Score	Top Accuracy	Validation Loss
RoBERTa	80%	0.59	0.82	0.80	0.67

Conclusion:

Model1(Finetuned Bert Base Uncased Model with Summary) was performing better on our validation data with lower validation loss and higher accuracy

3.6 Libraries Used

The models were implemented using the following libraries and frameworks:

- **PyTorch:** A deep learning framework used for training the models. PyTorch provides flexibility and is well-suited for research purposes.
- **Hugging Face Transformers and Tokenizers**: The Transformers library from Hugging Face made it easy to load pre-trained models (like DistilBERT, RoBERTa, T5, and BART) and fine-tune them for our specific tasks.
- **Scikit-learn:** This library was used for feature extraction and evaluation metrics
- **Pandas:** Used for data manipulation and preprocessing.

3.6. Summary of Model Choices and Performance

The models chosen for each task were specifically selected based on their performance, efficiency, and suitability for the respective NLP task. Each of the transformer models has unique advantages in handling specific aspects of the project, from understanding complex complaint narratives to generating concise summaries and classifying sentiments and responses.

Task	Models Used	Purpose
Complaint Classification	DistilBERT, RoBERTa	Classifying consumer complaints into predefined categories
Summarization	T5, BART	Extracting key information and generating concise complaint summaries
Sentiment Analysis	DistilBERT, RoBERTa	Classifying complaints based on sentiment (positive, negative, neutral)
Company Response Classification	BERT-BASE-UNCASED	Evaluating the timeliness and appropriateness of company responses

3.7. Conclusion

In this project, we leveraged the power of transformer-based models to tackle four different NLP tasks involving a consumer dataset. Each model was carefully chosen based on its capabilities and fine-tuned to suit the specific nature of the task. With fine-tuning and hyperparameter optimization, these models were able to achieve competitive performance in tasks such as complaint classification, summarization, sentiment analysis, and company response classification.

4. REFERENCES

In this section, we provide references for the background information, tools, libraries, and code repositories used in the project. This includes links to the models, datasets, research papers, and any code repositories we borrowed or referenced for this project.

4.1. Background Information and Models

1. Transformers:
    - Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need. In Advances in Neural Information Processing Systems (NeurIPS).
    - URL: https://arxiv.org/abs/1706.03762
2. DistilBERT:
    - Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper, and lighter. In arXiv.
    - URL: https://arxiv.org/abs/1910.01108
3. RoBERTa:
    - Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., & Lewis, M. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. In arXiv.
    - URL: https://arxiv.org/abs/1907.11692
4. BART:
    - Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Lewis, M., & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pretraining for Natural Language Generation, Translation, and Comprehension. In arXiv.
    - URL: https://arxiv.org/abs/1910.13461
5. T5 (Text-to-Text Transfer Transformer):
    - Raffel, C., Shazeer, N., Roberts, A., Lee, P. J., Narang, S., Matena, M., Zhou, Y., Figueroa, M., Liu, P. J., & Salzmann, T. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. In arXiv.
    - URL: https://arxiv.org/abs/1910.10683
6. Sentiment Analysis:
    - Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to Fine-Tune BERT for Text Classification? In arXiv.
    - URL: https://arxiv.org/abs/1905.05583
7. Data Augmentation Techniques for NLP:

- Wei, J., & Zou, K. (2019). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In arXiv.
- URL: https://arxiv.org/abs/1901.11196

4.2. Datasets

1. Consumer Complaint Dataset:
   - Consumer Financial Protection Bureau. (2021). Consumer Complaint Database.
   - URL: https://www.consumerfinance.gov/data-research/consumer-complaints/

4.3. Tools and Libraries

1. Hugging Face Transformers:
   - Hugging Face. (2020). Transformers: State-of-the-art Natural Language Processing for Pytorch, TensorFlow, and JAX.
   - URL: https://huggingface.co/transformers/
2. Scikit-learn:
   - Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. In Journal of Machine Learning Research.
   - URL: https://scikit-learn.org/
3. PyTorch:
   - Paszke, A., Gross, S., Massa, F., Lerer, A., Beaufour, F., & Chanan, G. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In NeurIPS.
   - URL: https://pytorch.org/

4.4. GitHub Repositories

- Complaint Classification and Sentiment Analysis Models:
  - URL: Project Github Link

4.5 Articles

https://medium.com/@saylibhavsar/analyzing-financial-complaints-with-nlp-7abc023333d1